

Discussion Papers

799

Sebastian von Engelhardt • Sushmita Swaminathan

**Open Source Software, Closed Source Software
or Both: Impacts on Industry Growth and the
Role of Intellectual Property Rights**

Berlin, June 2008

Opinions expressed in this paper are those of the author and do not necessarily reflect views of the institute.

IMPRESSUM

© DIW Berlin, 2008

DIW Berlin
German Institute for Economic Research
Mohrenstr. 58
10117 Berlin
Tel. +49 (30) 897 89-0
Fax +49 (30) 897 89-200
<http://www.diw.de>

ISSN print edition 1433-0210
ISSN electronic edition 1619-4535

Available for free downloading from the DIW Berlin website.

Discussion Papers of DIW Berlin are indexed in RePEc and SSRN.
Papers can be downloaded free of charge from the following websites:

http://www.diw.de/english/products/publications/discussion_papers/27539.html
<http://ideas.repec.org/s/diw/diwwpp.html>
http://papers.ssrn.com/sol3/JELJOUR_Results.cfm?form_name=journalbrowse&journal_id=1079991

Open Source Software, Closed Source Software or Both: Impacts on industry growth and the role of intellectual property rights[§]

Sebastian v. Engelhardt* Sushmita Swaminathan**

June 16, 2008

Abstract

There is considerable debate regarding the use of intellectual property rights (IPR) to spur innovation in the software industry. In this paper we focus on the choice of intellectual property right regimes and industry growth. We begin by developing a growth optimal mixture of open source and closed source software. This optimal scenario is then used as a basis to examine the co-existence of open and closed source software within various institutional frameworks ranging from no protection, copyright to patent protection. Such an analysis is beneficial as it enables an objective comparison of the three scenarios under the assumption that both copyrights and patents serve the purpose for which they were designed. Our analysis, based on the existence or absence of spillovers, confirms that a co-existence is growth optimal for the industry. Further, we find that the move from no protection to copyright protection increases the maximum growth rate. However, despite assuming properly functioning patents, the benefits of moving from copyright to patent protection are less clear.

Keywords: Intellectual Property Rights; Software; Open Source; Spillovers; Co-existence; Innovative Growth

JEL classification: L17; L86; O31; O34; O41; O43

[§]We would like to thank Pio Baake (DIW Berlin), Markus Pasche (FSU Jena), and Viktor Slavtchev (FSU Jena), as well as participants of the Annual Congress of the Society for Economic Research on Copyright Issues 2007 for valuable comments and suggestions.

*Friedrich-Schiller-University (FSU) Jena, Department of Economics, Carl-Zeiß-Str. 3, 07743 Jena, Germany, e-mail: Sebastian.Engelhardt@wiwi.uni-jena.de

**German Institute for Economic Research (DIW) Berlin, Mohrenstraße 58, 10117 Berlin, Germany, e-mail: sswaminathan@diw.de

Contents

1	Introduction	1
2	A brief introduction to software and IPR	2
3	The optimal mix of OSS and CSS	3
4	Three intellectual property right regimes	7
4.1	No protection	8
4.2	Copyright	10
4.3	Patents	11
5	Summary	15
A	Appendix	17
A.1	Spillovers, and growth-optimal input-share	17
A.2	Impacts of changes in α , β , or γ on maximum growth	18
A.3	Patents	19
A.3.1	The $d\gamma = 0$ case	19
A.3.2	The $d\gamma \neq 0$ case	21
	References	23

List of Figures

1	Share of Input (θ) for Maximum Growth	7
2	Relative Change of α and β and Input-Share	20
3	Case with $d\gamma = d\beta > 0$	21
4	Case with $d\gamma = -d\beta < 0$	22

List of Tables

1	Different Types of Software Licenses	8
---	--	---

1 Introduction

There is considerable debate regarding the use of intellectual property rights (IPR) to spur innovation in the software industry. Broadly speaking, copyrights and patents are generally the main IPR mechanisms used in the software industry. Specifically, questions regarding the “proper role of copyright” (Baseman et al., 1994) and the “economic impacts and policy implications” (Blind et al., 2005) of software patents are being extensively debated, as well as who uses software patents and why (Bessen and Hunt, 2004a)

The software industry has seen the emergence of two different types of technology sharing strategies (Jansen, 2006), namely proprietary and open source software (OSS). The main difference between the two technology sharing strategies lies in the provision of the source code. Proprietary software is sold only as binary code where access to the source code is prohibited.¹ Thus, some observers—and we follow this consideration—refer to proprietary software as closed source software (CSS).² In general, when the access to source code is “open”, it is referred to as OSS.³

Innovations in application software or operating systems are often sequential, complementary (Bessen and Maskin, 2000) and cumulative (Friedewald et al., 2002) where improvements are characterized as new expressions of existing knowledge. Thus, knowledge spillovers from competitors and collaborators alike can contribute to private investment and spur subsequent technical advances (Arrow, 1962; Jaffe, 1986). We take this a step further and suggest that the amount of spillovers is a direct consequence of the IPR regime in place.

We define the term *spillover* as the diffusion of information when the source code or the description of a programming solution is disclosed, the re-use and re-combination of (parts of) source code and learning by doing effects (Arrow, 1962; Jaffe, 1986). In this paper, we focus on the sum of two main spillover effects: First, knowledge that indirectly accrues from having access to the source code and improving upon it or creating something new.

¹Binary code refers to software that is only machine readable while source code refers to the human readable version of a computer program.

²The term ‘proprietary’ comes from the latin terms *proprietaryus* and *proprietas* and its legal meaning is ‘protected by copyrights’, and this is true for OSS as well.

³OSS is developed by communities including a broad range of participants from hobbyists to companies like IBM, HP, Sun Microsystems and OSS distributors like Red Hat or Novell’s SUSE.

Second, when existing source code is incorporated directly in to other source code, thereby, creating a recombinant effect. Thus, OSS and CSS essentially differ with respect to spillover effects.

The literature offers a rich variety of the advantages and disadvantages of patents and copyrights within the software industry and OSS and CSS individually. However, only a few papers have addressed the co-existence of OSS and CSS. We observe this co-existence in many software markets and software related markets including ERP, web server software and embedded software. Our main aim in this paper is to show that there exists a growth optimal co-existence and to examine this co-existence under varying IPR regimes. We start by developing a growth optimal base scenario. This benchmark scenario is then used to examine the growth optimal co-existence of OSS and CSS in the context of spillovers and innovation within varying IPR regimes. Such an analysis is beneficial as it enables an objective comparison of the three scenarios under the assumption that both copyrights and patents serve the purpose for which they were designed. In other words, they are free from the distorting effects which cast doubt on their effectiveness.

2 A brief introduction to software and IPR

Software is traditionally protected by copyright which protects the expression of an idea but not the idea itself. Thus, copyright prevents direct copying but allows the copying of the underlying idea or concept. In the United States, as of 1981, patent protection has also been extended to software. Patents also do not protect the underlying idea of an innovation but protect the new technical application of an idea. They are awarded for originality, novelty and non-obviousness in exchange for early disclosure of the innovation to society.

From a theoretical perspective, IPR are designed to create incentives to the innovator (ex-ante incentive) and to ensure that information is disclosed (ex-post efficiency) (Quah, 2003, p 19 ff). However, it has also been cited in the literature (Moser, 2005; Gallini and Scotchmer, 2001) that IPR need not be the only form of protection nor the main incentive for innovation. For example, Moser's (2005) empirical study on patents indicates that innovation occurs in the absence of patent protection. Thus, it could be said that patent law has limited impact on the number of innovations but tends to determine the kind of innovation that takes place and the direction of technical change.

Another example refers to [Cohen et al. \(2000\)](#) who empirically examine the relative importance of patents when compared to other mechanisms including secrecy, market dominance and lead time advantages.⁴ Thus, IPR are neither the only way to protect innovations nor is there an absence of innovation without IPR. Considering these findings, one could surmise that the main function of IPR is to minimize the (transaction) costs of protecting intellectual property. Hence, well defined IPR increase legal certainty and reduce transaction costs.⁵ Legal certainty has implications for innovative activity. Our focus rests on one of these implications: if well-defined IPR are guaranteed, economic agents are more willing and able to share innovative ideas and concepts through contract based relationships. Hence, the existence of well-defined IPR can change the way of producing (innovative) products

The rest of the paper is structured as follows: Section 3 describes the derivation of the growth optimal mixture of OSS and CSS. Section 4 examines the impact of varying legal frameworks i.e. ranging from no protection, copyright to patent protection. We conclude in Section 5.

3 The optimal mix of OSS and CSS

This section formally derives the growth optimal mixture of OSS and CSS and creates the basis for the analysis in the subsequent sections. We focus on one industry only—the software industry—with two sectors, A and B . A represents the OSS sector and B represents the CSS sector.

Let F be the total input stock available to produce the first copies of software. The *share of input* that is used to produce software of type A (OSS) is given by $\theta \cdot F$, with $0 \leq \theta \leq 1$. Therefore $(1 - \theta) \cdot F$ is used to produce software of type B . We assume, that the economy is in a stable equilibrium, thus $\theta(t)$ does not change over time given no change in exogenous parameters: $\theta(t) = \theta, \forall t$. Without loss of generality we normalize $F = 1$ so that the division of the given input stock is described by the value of θ .

We assume that the first-copy-production function for each sector (A or B) can be described with a simple linear function, i.e. the output is the

⁴For similar findings, see [Anton and Yao \(2004\)](#); [Arundel \(2001\)](#).

⁵On the other hand, IPR can increase transaction costs (i.e. information costs), as one has to inform whether e.g. a new application is already covered by somebody's IPR. For example this can potentially lead to problems described by the term 'patent thicket', we will come back to this later.

result of the input multiplied with a productivity measure, denoted by $p_A(t)$, or $p_B(t)$ respectively: $Y_A(t) = p_A(t)\theta$ and $Y_B(t) = p_B(t)(1 - \theta)$.

The productivity of sector A (B) is the product of a basic niveau-factor, denoted by a (b) and a spillover dependent part, denoted by $\sigma_A(t)$ ($\sigma_B(t)$), such that $p_A(t) = a(1 + \sigma_A(t))$ and $p_B(t) = b(1 + \sigma_B(t))$. At this point, we have make to clear, how we distinguish between ‘basic niveau’ factor and ‘spillover dependent part’: The basic niveau factor refers to the productivity that can be directly traced to the way the production process is organized. The spillover dependent part refers to information/knowledge spillover between agents (or units) that do not have any kind of contractual relationship. Notice, that our distinction between the spillover dependent part vs. the not spillover dependent part is pragmatic: The way a production process is organized determines the transaction costs, and the amount of re-use of code and concepts within this organization, as well as code and concept sharing between e.g. CSS firms that collaborate. These effects influence productivity, and we formalize this idea through the basic niveau factor. This can be clearly distinguished from external effects i.e. spillover effects. For example if a (CSS or OSS) firm uses (concepts from) given code that is protected by a public license, this is related to the spillover dependent part of productivity, as there is no direct organizational/contractual connection between the source code writer(s) and the firm.

The total output of the industry at time T (‘Today’) is given by

$$\begin{aligned} Y(T) &= Y_A(T) + Y_B(T) = a(1 + \sigma_A(T))\theta + b(1 + \sigma_B(T))(1 - \theta) \quad (1) \\ \text{with} \quad &a, b, \sigma_A(T), \sigma_B(T) \geq 0 \end{aligned}$$

The spillover dependent part of productivity is determined by the sector’s ability to benefit from the spillovers of each sector.

In the general case, every sector can benefit from inter- and intra-sectoral spillovers. To indicate the inter- and intra-sectoral spillovers, we label the inter-sectoral spillover effect of sector i on sector j with s^{ij} and similarly the intra-sectoral spillover effect of sector i with s^{ii} ($s^{ij}, s^{ii} \geq 0$).

It is easy to compute the division of F that *yields maximum growth*. We express the division of F by the input share of OSS (θ). Hence, the growth maximizing input share of OSS, i.e. the *optimal input share* is given by

$$\theta^* = \frac{1}{2} \cdot \frac{(as^{BA} + bs^{AB}) - 2 \cdot bs^{BB}}{(as^{BA} + bs^{AB}) - as^{AA} - bs^{BB}} \quad (2)$$

for an interior solution, and $\theta^* = 0$ or $\theta^* = 1$ otherwise (for details, see Appendix A.1). Hence, from $\theta \in [0, 1]$ we derive the following:

The boundary $\theta^* \geq 0$ implies

$$bs^{BB} \leq \frac{1}{2}(as^{BA} + bs^{AB}), \quad (3)$$

and $\theta^* \leq 1$ implies

$$as^{AA} \leq \frac{1}{2}(as^{BA} + bs^{AB}). \quad (4)$$

Thus, because of (3) and (4), an interior solution ($1 > \theta^* > 0$) always fulfills the second order condition (SOC) given by

$$as^{AA} < (as^{BA} + bs^{AB}) - bs^{BB}. \quad (5)$$

(see also figure 1). Thus, only within the $\frac{1}{2}(as^{BA} + bs^{AB})$ boundaries a mixture of OSS and CSS is growth optimal. The mixed area increases, if either A 's benefit resulting from the inter-sectoral spillovers (as^{BA}) or B 's benefit from the inter-sectoral spillovers (bs^{AB}) increases, or both. It is important to note, that the apportionment of as^{BA} and bs^{AB} does not play a role, only the sum ($as^{BA} + bs^{AB}$) determines the area where $1 > \theta > 0$ is optimal. This result is similar to findings on network theory concerning adapters: Regarding a two-network case, Church and King (1993) showed, that if adapters are costly to install, then it is optimal to install only one adapter that enables one network to benefit from the other. This implies, if adapters are not costly, then there is no need to install two adapters, it is only essential that there is an exchange. Thus spillovers have analogies with network externalities, which is not really surprising:

The growth of a sector (A or B) increases with an increase in a) its own share of input subsequently its share of output and the resulting market share and b) the other's sector share of input, as much as it can benefit from the other through inter-sectoral spillovers. The underlying logic is similar to the network case: If one now replaces the term 'growth' by 'utility', the term 'sector' by 'good', the term 'share of input' by 'installed base' and the term 'inter-sectoral spillovers' by 'adapter', we get the network adapter case: The utility of one good (A or B) increases with an increase in a) its own installed base and b) the other good's installed base, as much as it can benefit from the other's installed base by the adapter.

What both have in common is the fact, that there are feedback effects. If there are two groups (networks or sectors), they may be incompatible in the sense that the feedback effects are limited to each group. In such a case, connecting the groups is beneficial, and it might be sufficient that only one group can benefit from the feedback effects caused by the other. Hence with respect to spillovers and growth, only the sum $as^{BA} + bs^{AB}$ matters.

However, since $as^{BA} + bs^{AB}$ is the sum of the parts of productivity growth that depends on inter-sectoral spillovers, we refer to it in the following as the interdependent parts of productivity growth and denote it with γ , therefore $\gamma = (as^{BA} + bs^{AB})$. Additionally we will use $\alpha = as^{AA}$ and $\beta = bs^{BB}$ and call this the autonomous parts of productivity growth. This simplifies the first order condition (2) to

$$\theta^* = \frac{1}{2} \cdot \frac{\gamma - 2 \cdot \beta}{\gamma - \alpha - \beta} \quad (6)$$

and the boundaries (3) and (4) to

$$\beta \leq \frac{1}{2}\gamma, \quad (7)$$

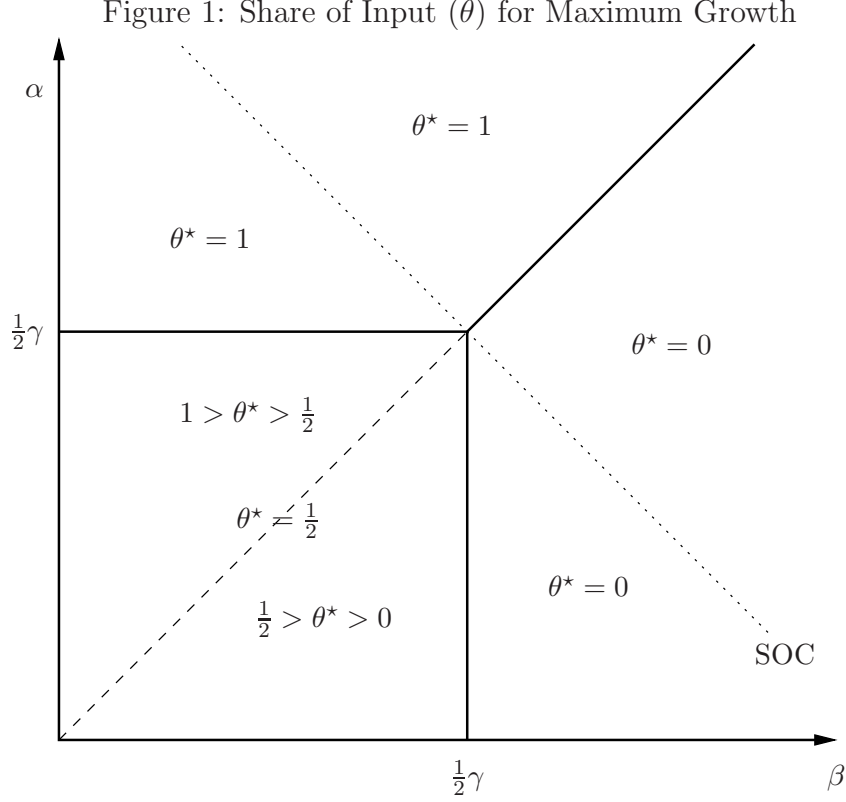
$$\alpha \leq \frac{1}{2}\gamma. \quad (8)$$

Figure 1 depicts (7) and (8), as well as (5). Additionally, figure 1 contains a $\theta^* = \frac{1}{2}$ -line, i.e. the line with $\alpha = \beta$ (i.e. $as^{AA} = bs^{BB}$). For any combination of α and β where the second order condition is *not* fulfilled, $\theta^* = 0$ if $\alpha < \beta$ (i.e. $as^{AA} < bs^{BB}$) and $\theta^* = 1$ if $\alpha > \beta$ (i.e. $as^{AA} > bs^{BB}$).

With (6) as well as the boundaries (7) and (8) one can easily compute maximum growth $\dot{Y}^*(T) = \dot{Y}(\theta^*, T)$ given by

$$\dot{Y}^*(T) = \dot{Y}(\theta^*, T) = \begin{cases} \alpha \cdot g(T) & \text{if } \theta^* = 1, \\ \beta \cdot g(T) & \text{if } \theta^* = 0, \\ \frac{\gamma^2 - 4\alpha\beta}{4(\gamma - \alpha - \beta)} \cdot g(T) & \text{else.} \end{cases} \quad (9)$$

To sum up: For some parameter-combinations, an OSS-CSS-mixture is growth optimal. The sum of the interdependent parts of productivity growth is what determines growth.



4 Three intellectual property right regimes

In the following sections we examine the effect of varying IPR on the software industry i.e. both the OSS and CSS sectors. We further examine the impact of IPR on spillover activity between the two sectors. In order to better gauge the level of spillovers that may occur through various license agreements, we start by describing the various licences. Software can broadly be distinguished into OSS and CSS, or rather into software with OS-licenses vs. software with CS-licenses. The different licenses have varying impacts on spillover activity. Although all types of software licenses are based on intellectual property law—i.e. copyright⁶—OS-licenses *includes* whereas non-OS-licenses *excludes*: In the case of CSS copyright is used (together with technical facilities i.e. copy protection) to protect exclusive ownership and

⁶It is often “misperceived [that OSS] remove[s] copyright protection. It is based on copyright principles.” (Gehring, 2006, p 62, 70).

therefore to prevent the disclosure of information, i.e. keep the source code ‘closed’ (see table 4). In the case of OSS there is general access to the source code, i.e. there is disclosure of information. Thus one could refer to OSS as the “private provision of a public good” (Johnson, 2002).

With respect to OSS-licenses, one can broadly distinguish between *public* and *viral* licenses (see table 4): *Public licenses*, like the Berkeley Software Distribution License (BSD), do not restrict the use of the software or the source code in any way, provided that credit is given to the original authors. This implies that software under such a license can not only be re-used and

Table 1: Different Types of Software Licenses

Type of License		General access to the source code?	Any restrictions in code usage (changing, use as input, etc.)?
CSS		No	– <i>no access</i> –
OSS	‘viral’ (GPL)	Yes	Yes (remain OSS)
	‘public’ (BSD)	Yes	No

re-combined but also redistributed under *any* preferred license, regardless of whether the code is changed and/or combined with other code. *Viral licenses* like the General Public License (GPL) differ in terms of alienation rights as the right to distribute is restricted. This means that software developed subsequently or derivative works based on some source code must be licensed as a whole under the same type of OSS license. This means that any redistributed software where (or parts of) viral OSS is included, becomes OSS as well. One could interpret the OSS licenses as being a contract to all, offering all users the rights to read, modify and use the software, while the possible constraints are only relevant upon redistribution.

The following three sections examine three cases: No protection, copy-right and software patents.

4.1 No protection

CSS is based on the principle of exclusive control over intellectual property. If there is no intellectual property law (IPL), CSS firms pursue a strategy

of hiding all their intellectual property in order to avoid any spillover activity. Therefore, in such a world, CSS would only be provided in binary code, protected through encryption or by using sophisticated technical copy protection minimizing spillover activity. Hence, in such a world *without any* IPR-protection, CSS-firms would tend to ‘hide everything’. So s^{BB} and s^{BA} are virtually zero, and therefore, we set $s^{BB} = s^{BA} = 0$ for this case.

With regard to the OSS case (sector A), the absence of IPR leads to only BSD like licenses. Thus, in this scenario, only A produces spillovers, from which both sectors benefit: s^{AA} and s^{AB} are both positive, thus $s^{AA} > 0$ and $s^{AB} > 0$.

With $s^{AA} > 0$, $s^{BB} = 0$, $s^{AB} > 0$, and $s^{BA} = 0$, the expressions (5), (7), and (8) yield

$$\theta^* = \begin{cases} \frac{1}{2} & \text{if } \alpha = 0 \\ \frac{1}{2} \cdot \frac{\gamma}{\gamma - \alpha} & \text{if } 0 < \alpha < \frac{1}{2}\gamma \\ 1 & \text{if } \alpha \geq \frac{1}{2}\gamma \end{cases} \quad (10)$$

with $\gamma = (0 + bs^{AB}) = bs^{AB}$.

Some authors argue, that “the open-source development method must consequently be classified as neither economically efficient nor effective.” (Kooths et al., 2003, p 64). However, *even if* the OSS sector has very low productivity levels, (i.e. a is virtually zero), a 50%-mixture of OSS and CSS is growth-optimal. As long as $\alpha < \frac{1}{2} \cdot \gamma$, a mixture is growth optimal, with the optimal share of OSS increasing from 50% to 100% when α is increasing up to $\frac{1}{2} \cdot \gamma$. For all $\alpha \geq \frac{1}{2} \cdot \gamma$, an input-share of 100% of OSS is optimal. Let us, for argument’s sake, assume that OSS might not be as productive as CSS i.e. $a < b$. However, considering the fact that BSD-like licenses enable both, CSS and OSS producers to use BSD-licensed source code (thus virtually $s^{AA} = s^{AB}$), it is at least not unlikely that $\alpha < \frac{1}{2} \cdot \gamma = \frac{1}{2} \cdot bs^{AB}$, $0 < bs^{AB}$ respectively.

To sum up: We obtain that a mixture of OSS and CSS is growth optimal in the case of no IPL. In other words: regarding figure 1, this situation would be indicated by a point at the α -axis, somewhat below the $\frac{1}{2} \cdot \gamma$ -frontier.

In the next subsections (4.2 and 4.3), we examine the impact of introducing copyright, and software patents respectively, on the maximum growth, i.e. its impact on the the optimal input share of OSS. Therefore, we focus on \dot{Y}^* and θ^* . Hence, notice, that *in the following we assume, that the realized θ always equals θ^* given by (6).*

4.2 Copyright

In this section we examine the effect of introducing copyrights and consider the impact on the co-existence of OSS and CSS and the impact on spillover activity, growth and innovation. These impacts enable us to draw some insights regarding copyright protection in the software industry.

Copyright protection could be beneficial to the CSS sector (sector B) as it could have a positive effect on the spillover-independent part of productivity b . Productivity increasing collaborations between CSS firms, i.e. contract based code sharing, are now possible for CSS firms. In a world *without* any IPR, cooperation between CSS firms based upon the sharing of source code (e.g. common development of software parts) does not occur or are at least not stable, as the lack of IPL provides no legal protection in the case of conflict. Thus, with the change from no protection to copyright protection, the effect on b would be positive: $db > 0$.

CSS firms can now exclusively protect their software by legal arrangements, creating no immediate need to hide everything. Recall that we assumed earlier that $s^{BB} = 0$. Thus, the impact of introducing copyright protection on the intra-sectoral spillovers of sector B can consequentially only be greater-than-or-equal to zero. If one follows our argument; in the case of IPR-protection, CSS firms allow for some spillovers (or: do not tend to hide anything anymore), then it is likely that with copyright-protection s^{BB} is positive, albeit small. This leads to $ds^{BB} > 0$.

The same holds with respect to the inter-sectoral spillovers from sector B , hence $ds^{BA} > 0$.

Regarding the OSS sector, the introduction of copyright law means, that GPL-like licenses are now possible. This implies a restriction of code usage due to the viral nature of GPL. Obviously this does not affect the economic agents of the OSS sector, as they produce OSS by definition regardless of whether it is under the BSD or GPL license. Therefore, we do not see any reason, why either the basic niveau factor or the the intra-sectoral spillovers of the OSS sector should change. Hence, introducing copyright leads to: $da = 0$ and $ds^{AA} = 0$.

With respect to the inter-sectoral spillovers from sector A , the introduction of copyright has a negative effect because of what we call the “replacement effect”: Some of the former BSD-projects now turn into GPL-protected ones. As GPL-like licenses are “viral” in nature, GPL-protected source code can not be used as input to produce CSS. Hence, due to a reduction in the

percentage of BSD projects, the CSS-sector benefits less from the OSS-sector. Therefore, the impact of introducing copyright on s^{AB} is negative: $ds^{AB} < 0$.

Thus, the impacts on the parameters of introducing copyright law are:

$$\begin{aligned} d\alpha = d[as^{AA}] &= da \cdot s^{AA} + a \cdot ds^{AA} + da \cdot ds^{AA} = 0 \\ d\beta = d[bs^{BB}] &= db \cdot s^{BB} + b \cdot ds^{BB} + db \cdot ds^{BB} > 0 \\ d[as^{BA}] &= da \cdot s^{BA} + a \cdot ds^{BA} + da \cdot ds^{BA} > 0 \\ d[bs^{AB}] &= db \cdot s^{AB} + b \cdot ds^{AB} + db \cdot ds^{AB} \leq 0 \\ &\Rightarrow d\gamma = d[as^{BA} + bs^{AB}] \leq 0 \end{aligned} \quad (11)$$

Therefore, introducing copyright *could* decrease the sum of interdependent parts of productivity growth, which *could* yield a decrease of growth. But it is important to notice, that $d[as^{BA} + bs^{AB}] < 0$ is a necessary but *not sufficient* condition for $d\dot{Y}^* < 0$. From (22) one derives the *sufficient* condition for $d\dot{Y}^*(T) < 0$ given by $d\beta(2\alpha - \gamma)^2 < -d\gamma(2\alpha - \gamma)(2\beta - \gamma)$. To give an impression of this condition, we look at the (unlikely) case, where the replacement effect is so strong, that it overcompensates the positive effect of ds^{BA} (i.e. $d\gamma = d[as^{BA} + bs^{AB}] < 0$) and also compensates the $ds^{BB} > 0$ (that is $-dbs^{AB} = das^{BA} + ds^{BB} \Leftrightarrow -d\gamma = d\beta$). Therefore we present in the following the condition for $-d\gamma = d\beta = 1$ ($\alpha, \beta < \frac{1}{2}\gamma$, $d\alpha = 0$):

$$d\dot{Y}^*(T) \begin{cases} < 0 & \text{only if } \alpha > \beta \\ \geq 0 & \text{else} \end{cases} \quad (12)$$

To sum up: Introducing copyright reduces maximum growth *only if* the decrease of s^{AB} (caused by the replacement effect) overcompensates all other effects, i.e. *only if*

$$\frac{\partial \dot{Y}^*}{\partial b} db + \frac{\partial \dot{Y}^*}{\partial s^{BB}} ds^{BB} + \frac{\partial \dot{Y}^*}{\partial s^{BA}} ds^{BA} < -\frac{\partial \dot{Y}^*}{\partial s^{AB}} ds^{AB}. \quad (13)$$

In all other cases, copyright leads to higher maximum growth, hence $d\dot{Y}^* \geq 0$.

4.3 Patents

In the next step, we examine the impact of introducing software patents on the co-existence of OSS and CSS. The literature suggests that if R&D is cumulative and sequential, then excessive protection impedes rather than promotes future innovation (Samuelson, 1990; Bessen and Maskin, 2000;

Scotchmer, 1991). Despite the de-merits of patent protection, we assume that patents work in the way they were designed to work.

Software has practical use only when it is interoperable particularly when considering that innovation is sequential i.e. based on the reuse and recombination of products, short innovation life cycles and network effects. In the case of copyrighted software, though rights are ensured to the original author and the expression, subsequent innovators have access to the underlying idea. However, while patents also do not protect the underlying idea, they protect the commercial use of a particular idea (Lemley and Cohen, 2001). The chances of identical expression by alternate innovators are lower than the chance of achieving identical applications resulting from a common idea. Hence, it might be said that patent protection limits innovative growth in the software industry.

Although the OSS sector does not actively use patents, this does not imply that software patents do not have an impact on the OSS sector. Software patents could lead to higher transaction costs for OSS developers due to the “patent thicket” (Shapiro, 2003), as information OSS programmers use and want to use, may be patent protected. A patent thicket refers to when a product involves many patents. Thus, this induces information costs and/or uncertainty and risk for OSS developers.⁷

Given the patent thicket exists, economic agents—from both the OSS and CSS sector—may face excessive transaction costs as they must negotiate with large numbers of patent holders (Bessen, 2004, p 1; Shapiro, 2003; Noel and Schankerman, 2006). This leads to a waste of resources. Additionally, there is a probability, that a started project has to be aborted as it violated someone’s patent. Again, this leads to a waste of resources (as produced source code can not be used anymore). Taking these effects together, there is a negative impact on the basic niveau factor of OSS ($da < 0$). Some authors (e.g. Mann) argue that this information problem of (software) patents is overemphasized. One can imagine that e.g. an online central database of patented software could solve the information problem. In such a scenario, software patents would not affect the spillover independent part of OSS productivity: $da = 0$. However, we regard it more likely that the introduction of software patents would lead to a decrease of a , thus, $da < 0$.

⁷For the discussion on software patents and the patent thicket problem see e.g. Mann (2005, 2004) vs. Bessen and Hunt (2004b); Bessen (2006)

As OSS developers do not use patents, no new licenses types emerge. Due to this, we assume no major change in s^{AA} and s^{AB} . In other words, neither in the ability of the OSS sector to benefit from spillovers from OSS sector, nor will the ability of the CSS sector to benefit from OSS spillovers be affected by the introduction of software patents: $ds^{AA} = ds^{AB} = 0$.

With respect to CSS firms, the negative impacts of software patents are in principle the same as for OSS. But, as CSS business models are based upon the exclusive use of code, there are also positive effects. An important question to consider here is; what is the qualitative difference between that of copyright protection and patent protection for software. Though an extensive analysis of this question is not within the scope of this paper, one obvious difference seems to be regarding enforcement. Innovators with a patent have more legal clout in ensuring the originality of their intellectual property in comparison to copyright protection.

Thus, while software patents increase transaction costs on the one hand e.g. through patent thickets, on the other they lower transaction costs, as they make it easier to enforce rights. By making enforcement easier, firms are more willing to establish interfirm collaborations. As we want to examine the effects of patents in the case they work as they should i.e. software patents are a good thing for business models based upon the exclusive ownership of source code, we assume the following:⁸ Even if one takes into account the negative effects of software patents for CSS developers, the positive effects at least balance this, such that $db \geq 0$.

Software patent applications cannot contain complete information as this would mean revealing the source code. It could be argued that descriptions provided in patent applications could lead to positive spillovers. However, descriptions given in exchange for patent protection are often not clear and precise enough for it to be beneficial to the others (Levin, 1988; Cohen et al., 2002). This could result in a duplication of research efforts whereas in a setting with interdependent spillovers, wasteful expenditures on R&D could be reduced (Blind et al., 2005). Alternatively, spillovers do occur despite such protection through the trading and sharing of knowledge, labor mobility and possibility to reverse engineer. Crampes and Langinier (2005) found, under certain conditions firms chose not to renew their patents to prevent

⁸Notice, that we do not claim, that this is a realistic assumption. But if we assume, that software patents do not work, i.e. that they are not beneficial for CSS firms, then the result would imply that introducing software patents leads to an inferior situation.

information from entering the market. Without explicitly specifying the relationship between this positive amount of information and spillover effects, it is reasonable to assume a positive effect, especially if one wants to assume properly functioning patents, which is our intention. Thus: $ds^{BB} > 0$ and $ds^{BA} > 0$.

To sum up, the expected impact on the parameters of introducing software patents are:

$$\begin{aligned}
 d\alpha = d[as^{AA}] &= da \cdot s^{AA} + a \cdot ds^{AA} + da \cdot ds^{AA} < 0 \\
 d\beta = d[bs^{BB}] &= db \cdot s^{BB} + b \cdot ds^{BB} + db \cdot ds^{BB} > 0 \\
 d[as^{BA}] &= da \cdot s^{BA} + a \cdot ds^{BA} + da \cdot ds^{BA} \leq 0 \\
 d[bs^{AB}] &= db \cdot s^{AB} + b \cdot ds^{AB} + db \cdot ds^{AB} > 0 \\
 &\Rightarrow d\gamma = d[as^{BA} + bs^{AB}] \leq 0
 \end{aligned} \tag{14}$$

Due to $d\gamma \leq 0$ we can identify three possible scenarios:

- $d\gamma = 0$, in this case, the impact of introducing software patents on \dot{Y}^* depends on the relative impact of $d\alpha$ and $d\beta$ only. It turns out, that $d\dot{Y}^*(T) > 0$ if $d\alpha(2\beta - \gamma)^2 + d\beta(2\alpha - \gamma)^2 > 0$. Let us express the relative change of α and β with k , such that $d\alpha = -k \cdot d\beta$. This yields the following:

For any given relative reverse change of α and β —i.e. for any given $k > 0$ —there is an area of (α, β) -combinations, where the maximum growth rate *increases* because of $-d\alpha < k \cdot d\beta$, i.e. the decrease of α is overcompensated by the increase of β (for details, see Appendix A.3.1).

Hence, in case of $d\gamma = 0$ it depends on θ^* , whether introducing software patents more likely increases or decreases maximum growth.

- $d\gamma \neq 0$, in this case, $d\gamma$ either weakens or strengthens the condition (for details, see Appendix A.3.2). Hence, in case of $d\gamma > 0$ software patents are more likely to increase growth, as the condition for $d\dot{Y}^*(T) > 0$ now is

$$d\beta(2\alpha - \gamma)^2 + d\gamma(2\alpha - \gamma)(2\beta - \gamma) > -d\alpha(2\beta - \gamma)^2, \tag{15}$$

while in case of $d\gamma < 0$ this is

$$d\beta(2\alpha - \gamma)^2 > -[d\gamma(2\alpha - \gamma)(2\beta - \gamma) + d\alpha(2\beta - \gamma)^2]. \tag{16}$$

To sum up: Although our assumptions imply the existence and impact of properly functioning software patents, it is not unlikely, that the introduction of software patents *decreases* maximum growth. Given that the economy's actual input-share θ is equal to the growth optimal θ^* , one can say, that introducing software patents more likely increases growth, the smaller the input share of OSS, i.e. the higher the share of CSS.

5 Summary

In this paper we examine the impact of IPR on the co-existence of OSS and CSS and how it affects growth of the software industry. In order to objectively assess copyright and patent mechanisms in the software industry, this paper developed a formal maximum growth mixture of OSS and CSS. The main aim in doing so was to create a benchmark scenario based on the assumption that both copyrights and patents in the software market serve the purpose they were designed for. In other words, they are free from the distorting effects which cast doubt on their effectiveness.

Our main findings can be summarized as follows:

- (a) It is possible to have a mixture of OSS and CSS that is growth-optimal. In other words, we show that there is a co-existence and observe how IPL affects this co-existence using the spillover argument.
- (b) We further find that it is not important for the benefits of intra-spillover activity to be equal. Hence, only the sum of interdependent parts of productivity growth matters. This result is similar to the function of adapters in networks and the spillovers can be interpreted as being network externalities. This means that a policy measure that increases one of the interdependent parts of productivity can yield a superior situation even if this implies a decrease for the other part. The important thing is that the sum of interdependent parts increases overall.
- (c) We find that changing from non-protection to copyright protection increases the maximum growth rate.
- (d) Although the existing economic literature points out that the use of patent protection for software is still fraught with much debate, we build

our framework on the assumption of properly functioning patents. However, we still observe scenarios where $d\dot{Y}^* > 0$ and where $d\dot{Y}^* < 0$. Thus, even in such a case, introducing software patents can lead to less growth.

In this context, public policy should perhaps focus on measures that increase either the basic niveau of productivity within the sectors or on the interdependent parts of productivity growth (or both) without affecting the autonomous part of productivity growth negatively. Despite modelling IPR in a favorable way, the benefits of patents still remain unclear. In the context of software, it might make greater sense to reduce the amount of breadth and length of a patent based on the amount of disclosure within the patent application.

The co-existence of OSS and CSS is still an area in the literature that needs further examination. However, though it provides interesting insights towards future research, it is to be noted that the analysis above is just a formalization of arguments and not a model in itself. Thus, further theoretical as well as empirical research examining the co-existence and interaction of OSS and CSS in the context of IPR is required.

A Appendix

A.1 Spillovers, and growth-optimal input-share

The spillovers of each sector are approximated by the discounted cumulated input of this sector, given by $\int_0^T v(T, t) \cdot \theta dt$, and $\int_0^T v(T, t) \cdot (1 - \theta) dt$ respectively. The function $v(T, t) = v(T - t)$ is a given discount function such that

$$1 \geq v(T, t) \geq 0 \forall t, \quad \lim_{(T-t) \rightarrow \infty} v(T, t) = 0, \quad \lim_{(T-t) \rightarrow 0} v(T, t) = 1.$$

We also introduce $g(T)$, which we will need for the following analysis:

$$g(T) = \frac{\partial \int_0^T v(T, t)}{\partial T} = \int_0^T \frac{\partial v(T, t)}{\partial T} dt + v(T, T) \leq 1.$$

As already mentioned in the text, we denote by a^{AA} (s^{BB}) the intra-sectoral spillovers of A (B), and with a^{AB} (s^{BA}) we denote the inter-sectoral spillovers caused by A (B) and being beneficial for B (A). Thus, the spillover dependent part of productivity is described as follows:

$$\begin{aligned} \sigma_A(T) &= s^{AA} \int_0^T v(T, t) \theta dt + s^{BA} \int_0^T v(T, t) (1 - \theta) dt \\ \frac{d\sigma_A}{dt} &= s^{AA} \theta g(T) + s^{BA} (1 - \theta) g(T) \end{aligned} \tag{17}$$

$$\begin{aligned} \sigma_B(T) &= s^{BB} \int_0^T v(T, t) (1 - \theta) dt + s^{AB} \int_0^T v(T, t) \theta dt \\ \frac{d\sigma_B}{dt} &= s^{BB} (1 - \theta) g(T) + s^{AB} \theta g(T) \end{aligned} \tag{18}$$

With (1), (17) and (18) it is easy to compute the growth of output ($\dot{Y} = dY/dt$).

As we want to know the division of F that yields maximum growth, we compute the optimal θ^* that implies maximum \dot{Y} , thus our optimizing problem is given by:

$$\max_{\theta \in [0, 1]} \dot{Y}(T) = g(T) (\theta^2 \cdot a s^{AA} + (1 - \theta) b s^{BB} + \theta(1 - \theta)(a s^{BA} + b s^{AB}))$$

And the First order condition yields

$$\theta^* = \frac{1}{2} \cdot \frac{(as^{BA} + bs^{AB}) - 2 \cdot bs^{BB}}{(as^{BA} + bs^{AB}) - as^{AA} - bs^{BB}}$$

for an interior solution, and $\theta^* = 0$ or $\theta^* = 1$ otherwise. The second order condition (SOC) of $\max_{\theta \in [0,1]} \dot{Y}$ yields

$$as^{AA} < (as^{BA} + bs^{AB}) - bs^{BB}. \quad (19)$$

And the boundary $\theta^* \geq 0$ implies

$$bs^{BB} \leq \frac{1}{2}(as^{BA} + bs^{AB}), \quad (20)$$

and $\theta^* \leq 1$ implies

$$as^{AA} \leq \frac{1}{2}(as^{BA} + bs^{AB}). \quad (21)$$

Thus, because of (20) and (21), an interior solution ($1 > \theta^* > 0$) always fulfills the SOC (19)

A.2 Impacts of changes in α , β , or γ on maximum growth

As already mentioned above, growth is described by the following function: $\dot{Y}(T) = g(T) (\theta^2 \cdot as^{AA} + (1 - \theta)bs^{BB} + \theta(1 - \theta)(as^{BA} + bs^{AB}))$.

With

$$\theta^* = \frac{1}{2} \cdot \frac{\gamma - 2 \cdot \beta}{\gamma - \alpha - \beta},$$

and the boundaries $\beta \leq \frac{1}{2}\gamma$ and $\alpha \leq \frac{1}{2}\gamma$, one derives maximum growth $\dot{Y}^*(T) = \dot{Y}(\theta^*, T)$ given by

$$\dot{Y}^*(T) = \dot{Y}(\theta^*, T) = \begin{cases} \alpha \cdot g(T) & \text{if } \theta^* = 1, \\ \beta \cdot g(T) & \text{if } \theta^* = 0, \\ \frac{\gamma^2 - 4\alpha\beta}{4(\gamma - \alpha - \beta)} \cdot g(T) & \text{else.} \end{cases}$$

Based on this, one can derive

$$\frac{\partial \dot{Y}^*(T)}{\partial \alpha} = \begin{cases} \frac{(2\beta - \gamma)^2}{4(\alpha + \beta - \gamma)^2} \cdot g(T) > 0 & \text{if } \alpha, \beta < \frac{1}{2}\gamma, \\ g(T) > 0 & \text{else} \end{cases}$$

and

$$\frac{\partial \dot{Y}^*(T)}{\partial \beta} = \begin{cases} \frac{(2\alpha - \gamma)^2}{4(\alpha + \beta - \gamma)^2} \cdot g(T) > 0 & \text{if } \alpha, \beta < \frac{1}{2}\gamma, \\ g(T) > 0 & \text{else} \end{cases}$$

as well as

$$\frac{\partial \dot{Y}^*(T)}{\partial \gamma} = \begin{cases} \frac{(2\alpha - \gamma)(2\beta - \gamma)}{4(\alpha + \beta - \gamma)^2} \cdot g(T) > 0 & \text{if } \alpha, \beta < \frac{1}{2}\gamma, \\ 0 & \text{else} \end{cases}.$$

A simultaneous change of α , β and γ leads to a change of $\dot{Y}^*(T)$ as given by the total derivative

$$\begin{aligned} d\dot{Y}^*(T) &= \frac{d\alpha(2\beta - \gamma)^2 + d\beta(2\alpha - \gamma)^2 + d\gamma(2\alpha - \gamma)(2\beta - \gamma)}{4(\alpha + \beta - \gamma)^2} \cdot g(T), \\ \Rightarrow d\dot{Y}^*(T) > 0 \quad \forall d\alpha(2\beta - \gamma)^2 + d\beta(2\alpha - \gamma)^2 + d\gamma(2\alpha - \gamma)(2\beta - \gamma) > 0 \end{aligned}$$

A.3 Patents

A.3.1 The $d\gamma = 0$ case

In this case (22) simplifies to $d\dot{Y}^*(T) > 0 \quad \forall d\alpha(2\beta - \gamma)^2 + d\beta(2\alpha - \gamma)^2 > 0$. With $d\alpha = -k \cdot d\beta$ we get

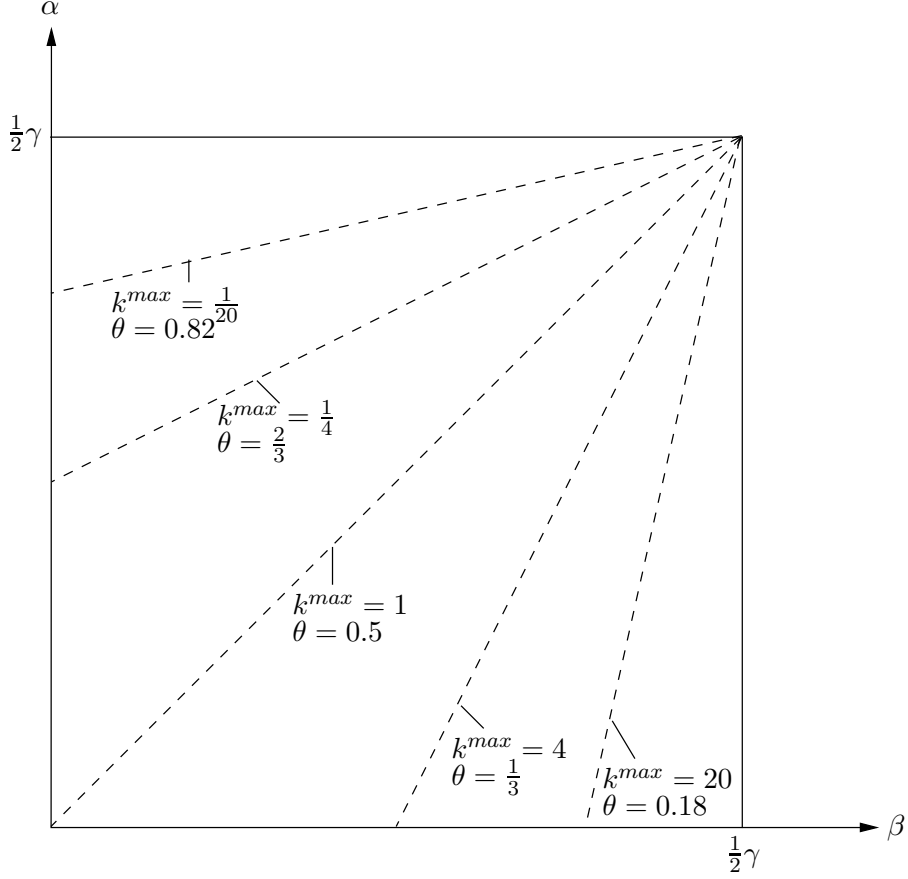
$$\forall d\alpha = -k \cdot d\beta, k > 0, d\beta > 0 : d\dot{Y}^* \geq 0 \iff k \leq \frac{(2\alpha - \gamma)^2}{(2\beta - \gamma)^2}$$

Thus, for any given k^{max} one can compute boundaries where $d\dot{Y}^*(k^{max}) = 0$ and hence $\forall k < k^{max} \quad d\dot{Y}^*(k) > 0$:

$$\alpha = \frac{1}{2}\gamma - \frac{1}{2}\sqrt{k^{max}(\gamma - 2\beta)^2}. \quad (22)$$

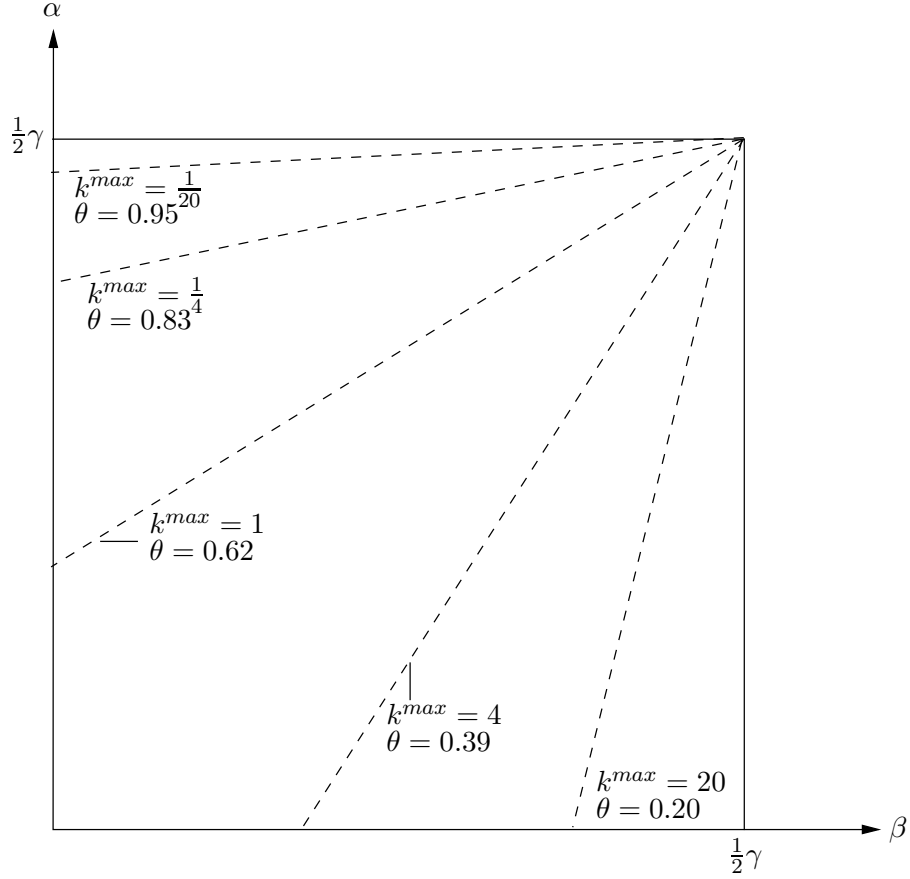
Using this to replace α in (6) we get:

$$\theta^* = \frac{1}{2} \cdot \frac{\gamma - 2 \cdot \beta}{\gamma - 2 \cdot \beta - \sqrt{k^{max}(\gamma - 2\beta)^2}}. \quad (23)$$

Figure 2: Relative Change of α and β and Input-Share

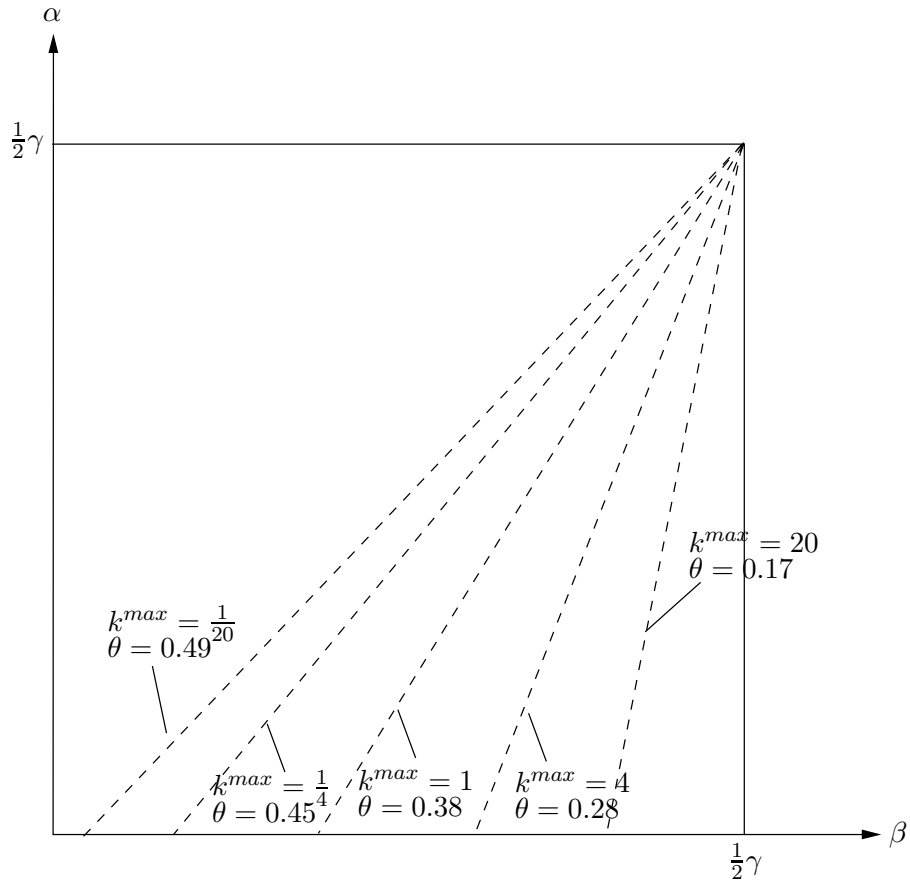
This implies the following: Because of the assumption that θ always equals θ^* (p 9) we can compute for every given input-share the condition which guarantees, that introducing software patents leads to a higher maximum growth.

Figure 2 depicts some examples. As one can see, the general result is, that the higher the share of OSS, the stronger the condition, i.e. the higher θ the smaller the k^{max} . A smaller k^{max} means that either positive impact on the CSS sector has to be higher, or the negative impact on OSS has to be smaller or both. For example in case of $k = 1$, i.e. $d\alpha = -d\beta$ the condition for $d\dot{Y}^* > 0$ is $\theta \leq 0.5$.

A.3.2 The $d\gamma \neq 0$ caseFigure 3: Case with $d\gamma = d\beta > 0$ 

In order to provide a visual impression of the impact of $d\gamma \neq 0$, figure 3 and figure 4 depict the boundaries for the same k -values as in figure 2. The figure 3 represents the situation for $d\gamma = d\beta > 0$, while the figure 4 represents the $d\gamma = -d\beta < 0$ case.

Thus, if one compares the situation with no change of the interdependent parts of productivity growth (figure 2) with figure 3, one can see how $d\gamma = d\beta > 0$ weakens the condition, as i.e. in the case of $k = 1$ introducing software patents would increase growth only if $\theta \leq 0.62$; while in the $d\gamma = 0$ case, it was $\theta \leq 0.5$.

Figure 4: Case with $d\gamma = -d\beta < 0$ 

Additionally, if one compares figure 2 with figure 4, one can easily see how $d\gamma = -d\beta < 0$ strengthens the condition such that in case of $k = 1$ introducing software patents would increase growth only if $\theta \leq 0.38$.

References

- Anton, J. J. and D. A. Yao (2004, Spring). Little patents and big secrets: Managing intellectual property. *RAND Journal of Economics* 35(1), 1–22.
- Arrow, K. (1962). Economic welfare and the allocation of resources for inventions. In R. Nelson (Ed.), *The Rate and Direction of Inventive Activity*, pp. 609–625. Princeton University Press.
- Arundel, A. (2001, April). The relative effectiveness of patents and secrecy for appropriation. *Research Policy* 30(4), 611–624.
- Baseman, K. C., F. R. Warren-Boulton, and G. A. Woroch (1994). The economics of intellectual property protection for software. the proper role for copyright. Paper, Economics Working Paper Archive at WUSTL, Washington.
- Bessen, J. (2004). Patent thickets: Strategic patenting of complex technologies. Working Papers 0401, Research on Innovation.
- Bessen, J. (2006). A comment on do patents facilitate financing in the software industry?. Working Papers 0601, Research on Innovation.
- Bessen, J. and R. M. Hunt (2004a). An empirical look at software patents. Working Papers 03-17, Federal Reserve Bank of Philadelphia.
- Bessen, J. and R. M. Hunt (2004b). The software patent experiment. In OECD (Ed.), *Patents, Innovation And Economic Performance*. Paris: OECD.
- Bessen, J. E. and E. S. Maskin (2000, 1). Sequential innovation, patents, and imitation. MIT Dept. of Economics Working Paper 00-01.
- Blind, K., J. Edler, and M. Friedewald (2005). *Software Patents – Economic Impacts And Policy Implications*. Cheltenham [u.a.]: Elgar.
- Church, J. and I. King (1993). Bilingualism and network externalities. *Canadian Journal of Economics* 26(2), 337–345.
- Cohen, W. M., A. Goto, A. Nagata, R. R. Nelson, and J. P. Walsh (2002, December). R&D spillovers, patents and the incentives to innovate in japan and the united states. *Research Policy* 31(8-9), 1349–1367.

- Cohen, W. M., R. R. Nelson, and J. P. Walsh (2000, February). Protecting their intellectual assets: Appropriability conditions and why u.s. manufacturing firms patent (or not). NBER Working Papers 7552, National Bureau of Economic Research, Inc.
- Crampes, C. and C. Langinier (2005, March). Are intellectual property rights detrimental to innovation? Staff General Research Papers 12267, Iowa State University, Department of Economics.
- Friedewald, M., K. Blind, and J. Edler (2002). Innovationstätigkeit der deutschen softwareindustrie, die. *Wirtschaftsinformatik* 44(2), 151–161.
- Gallini, N. and S. Scotchmer (2001, AUG.). Intellectual property. when is it the best incentive system? Working Paper E01 303, Economics Department, University of California, Berkeley.
- Gehring, R. A. (2006, March). The institutionalization of open source. *Poiesis & Praxis: International Journal Of Technology Assessment And Ethics Of Science* 4(1), 54–73.
- Jaffe, A. B. (1986, December). Technological opportunity and spillovers of R&D: Evidence from firms' patents, profits, and market value. *American Economic Review* 76(5), 984–1001.
- Jansen, J. (2006). Share to scare: Technology sharing incentives in the absence of intellectual property rights. Working paper.
- Johnson, J. P. (2002, December). Open source software: Private provision of a public good. *Journal of Economics & Management Strategy* 11(4), 637–662.
- Kooths, S., M. Langenfurth, and N. Kalwey (2003). *Open-Source Software: An Economic Assessment*, Volume 4 of *MICE Economic Research Studies*. Münster: Muenster Institute For Computational Economics.
- Lemley, M. A. and J. E. Cohen (2001). Patent scope and innovation in the software industry. *California Law Review* (89), 1–57.
- Levin, R. C. (1988, May). Appropriability, R&D spending, and technological performance. *American Economic Review* 78(2), 424–28.

- Mann, R. (2004). The myth of the software patent thicket. bepress Legal Series. Working Paper 183.
- Mann, R. (2005). Do patents facilitate financing in the software industry? *Texas Law Review* 38(4), 961–1030.
- Moser, P. (2005, September). How do patent laws influence innovation? evidence from nineteenth-century world’s fairs. *American Economic Review* 95(4), 1214–1236.
- Noel, M. D. and M. Schankerman (2006, May). Strategic patenting and software innovation. CEPR Discussion Papers 5701, C.E.P.R. Discussion Papers.
- Quah, D. (2003). Digital goods and the new economy. CEP Discussion Papers 563, London School of Economics, London.
- Samuelson, P. (1990). Benson revisited: The case against patent protection for algorithms and other computer program-related inventions. *Emory Law Journal* 39, 1025–1154.
- Scotchmer, S. (1991, Winter). Standing on the shoulders of giants: Cumulative research and the patent law. *Journal of Economic Perspectives* 5(1), 29–41.
- Shapiro, C. (2003, March). Navigating the patent thicket: Cross licenses, patent pools, and standard-setting. Law and Economics 0303005, Econ-WPA.